

Parser and Interpreter development for the  
Fuzzy Behavior Description Language in  
MATLAB  
Thesis outline and specifications

Nagy Balázs  
EIO1RQ, balazsnagy220@gmail.com

2023. december 13.

# 1. Introduction

## 1.1. Fuzzy behavior

A behavior-beased system (BBS) is complex hierarchical structure that tries to produce output in the same manner as a conscious being would react to certain inputs; as the name implies simulating its behavior. They offer interesting applications, mainly in robotics and embedded systems, this is especially true for BBS that use fuzzy interpolation in order to calculate their output, a method by which we can achieve more natural results that are closer to actual behavioral responses. These systems operate under the limitations of certain rules called rule-bases. We describe these rules of a given system using a descriptive language (FBDL), which is a high level programming language that allows for the configuration of the system even by those who may not be familiar with its inner workings.

## 1.2. Goals

The aim of the thesis and the subsequent software is to provide an effective and easy to use language processor for the aforementioned FBDL. Furthermore the program also calculates the supposed output via fuzzy rule interpolation using the predefined rule-bases. The entirety of the code shall be implemented using MATLAB and GNU/Octave compliant and compatible functions and syntax. As a result the software framework will be available for use in MATLAB and or GNU/Octave projects alike.

# 2. Specifications

The language of implementation was chosen due to demand as most of the projects using fuzzy behavior were written in MATLAB by the researchers at the University of Miskolc.

## 2.1. Processing the input

The very first step in analyzing the description of the rule-bases provided by the user and categorizing it as certain predefined tokens. The tokenizer or lexical analyzer conducts this task and forwards a constant stream of tokens to the parser for the next step in which the parser builds a syntax tree for

later evaluation of the individual statements. And finally the statements will serve as inputs to a fuzzy state machine controller (Fuzzy Rule Interpolation Behavior Engine or FRIBE) that evaluates them and produces a final state as a result. Any errors detected during the process are reported to the standard error along with the type of the error, an error message and the location where the fault occurred.

### **3. Evaluation**

During the evaluation of inputs extra attention must be dedicated to the fuzzy state machine and the operation of the FRIBE itself and how the different behavior components influence the final output.

### **4. Documentation**

All documentation is done with the use of git version control and source code is available on GitHub as well. The same terms apply to any technical documentation that might revolve around the framework in the future.